

PATENT APPLICATION

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re application of:

NGUYEN, Lynh

Appln. No.: 09/750,475

Confirmation No.: 7832

Filed: December 28, 2000

Docket No: STL919990134US3/A8644

Group Art Unit: 2452

Examiner: CHANKONG, DOHM

For: DATA SOURCE INTERFACE ENHANCED ERROR RECOVERY

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Mail Stop Appeal Briefs--Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 41.37, Appellant submits the following:

Table of Contents

I.	REAL PARTY IN INTEREST	2
II.	SUMMARY OF THE CLAIMED SUBJECT MATTER	3
III.	ARGUMENT	6
	CLAIMS APPENDIX.....	12

I. REAL PARTY IN INTEREST

The real party in interest in the appeal is International Business Machines, Corporation (“IBM”) of Armonk, New York, the assignee.

II. SUMMARY OF THE CLAIMED SUBJECT MATTER

Appellant's invention as recited in, for example, independent claims 1, 8, and 15, is related to a method, computer readable medium, and system for automatically re-establishing a connection to a data source accessible by a plurality of remote applications.

Claim 1

A method for automatically re-establishing a connection to a data source (see FIG. 7, data source 210) accessible by a plurality of remote applications (see FIG. 7, browser 202), the method comprising providing at least one interface module configured to interface with a remote application (see FIG. 7, interface module 208, browser 202; page 12, line 21 to page 13, line 6), providing at least one port module configured to interface between the interface module and the data source (see FIG. 7, port module 222, data source 210; page 13, 20-24), providing a connection manager to facilitate the interface between the interface module and the port module (see FIG. 7, connection manager 220; page 13, lines 14-20), detecting unavailability of the data source, by the at least one port module, in response to an initial request for the data source by the remote application (see page 21, lines 7-13); dynamically detecting availability of the data source, by the at least one port module, in response to a subsequent request for the data source (see page 21, lines 13-23); and re-connecting the data source to the remote application in response to the subsequent request (see page 21, lines 13-23), wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source (page 21, lines 17-19), reestablishes a connection with the data source, and reconnects the remote application to the data source directly communicating with the remote application (page

21, lines 19-23), and wherein the at least one port module bypasses the connection manager in the subsequent request (page 21, line 19 –page 22, line 9).

Claim 8

A non-transitory computer readable medium having stored thereon computer executable instructions for performing a method for connecting a plurality of remote applications with a data source (see page 8, lines 14-18), the method comprising providing at least one interface module configured to interface with a remote application (see FIG. 7, interface module 208, browser 202; page 12, line 21 to page 13, line 6), providing at least one port module to interface between the interface module and the data source (see FIG. 7, port module 222, data source 210; page 13, 20-24), providing a connection manager to facilitate the interface between the interface module and the port module (see FIG. 7, connection manager 220; page 13, lines 14-20), detecting unavailability of the data source, by the at least one port module, in response to an initial request for the data source by the remote application (see page 21, lines 7-13); dynamically detecting availability of the data source, by the at least one port module, in response to a subsequent request for the data source(see page 21, lines 13-23); and re-connecting the data source to the remote application in response to the subsequent request (see page 21, lines 13-23), wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source (page 21, lines 17-19), reestablishes a connection with the data source, and reconnects the remote application to the data source directly communicating with the remote application (page 21, lines 19-23), and wherein the at least one port module bypasses the connection manager in the subsequent request (page 21, line 19 –page 22, line 9).

Claim 15

A system for connecting a plurality of remote applications with a data source (see FIG. 7), the system comprising an interface module (see FIG. 7, interface module 208) configured to interface with a remote application (see FIG. 7, browser 202), a port module (see FIG. 7, port module 222) configured to interface between the interface module and the data source (see FIG. 7, data source 210), and a connection manager module (see FIG. 7, connection manager 220) configured to facilitate an interface between the interface module and the port module, wherein the port module further configured to detect unavailability of the data source in response to an initial request for the data source by the remote application (see page 21, lines 7-13), to dynamically detect availability of the data source in response to a subsequent request for the data source (see page 21, lines 13-23), and to re-connect the data source to the remote application in response to the subsequent request (see page 21, lines 13-23), wherein the port module sends an error message to the interface module indicating the unavailability of the data source (page 21, lines 17-19), reestablishes a connection with the data source, and reconnects the remote application to the data source by directly communicating with the remote application (page 21, lines 19-23), and wherein the port module bypasses the connection manager in the subsequent request (page 21, line 19 –page 22, line 9).

III. ARGUMENT

A. CLAIMS 1, 6-8, 13-15, 18-22 and 24

Claims 1, 6-8, 13-15, 18-22 and 24 are rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Yousefi'zadeh, (U.S. Patent No. 6,950,848; hereinafter "Yousefi'zadeh") in view of Guenthner et al, (U.S. Patent No. 6,134,588; hereinafter "Guenthner"), in further view of Albert et al., (U.S. Patent No. 6,549,516; hereinafter "Albert").

Applicant respectfully submits that the claimed invention would not have been rendered obvious in view of the combinations of the cited prior art.

1. Claims 1, 8 and 15

Appellant respectfully requests the Board of Patent Appeal and Interferences (Board) to reverse the rejection of independent claims 1, 8 and 15 at least because there is no reasonable combination of Yousefi'zadeh, Guenthner and Albert that would meet all of the recitation of claims 1, 8 and 15.

The combination of Yousefi'zadeh, Guenthner and Albert does not teach or suggest that at least one port module bypasses the connection manager in the subsequent request

Claim 1 recites, *inter alia*, "wherein the at least one port module bypasses the connection manager in the subsequent request." Claims 8 and 15 (in some variation) recite features similar to claim 1. According to these features, at least one port module bypasses the connection manager in the subsequent request for the data source. That is, the connection manager is bypassed in the subsequent request for the data source.

In page 4 of the Final Office Action dated December 28, 2011 (Office Action), the Examiner acknowledges that Yousefi'zadeh does not teach or suggest the features of "wherein the at least one port module bypasses the connection manager in the subsequent request," but cites column 13, lines 4-8 of Albert for disclosing these features (see page 4, line 4 of the Office

Action). Appellant respectfully disagrees for at least the following reasons because even if the teachings of the references were combined as the Examiner asserts, the combination would not satisfy all the limitations of claims 1, 8 and 15.

Albert relates to providing network services such as load balancing, packet filtering or Network Address Translation (NAT). Albert discloses a system that includes a service manager that determines how a network service is provided for a data flow and sends instructions to routers that detect packets for the data flow when such packets are actually detected by the routers (column 2, lines 12-20). In the cited portion of the reference, Albert discloses that if future packets in either flow sent from the client or host match the affinity key in one of the fixed affinities and are handled by the forwarding agent in accordance with the fixed affinity, then it is not necessary to forward the packet to the service manager (column 13, lines 4-8). Here, Albert merely discloses that it is not necessary for a packet having a matching affinity key within a data flow to be forwarded to the service manager. However, Albert is not at all concerned with the subsequent request for the data source, as recited in claim 1.

Moreover, in the Advisory Action, the Examiner cites column 6, lines 60-63 of Albert in further support for his contention that Albert discloses the features of “wherein the at least one port module bypasses the connection manager in the subsequent request for the data source.”

However, contrary to the Examiner’s assertion, in column 6, lines 60-63, Albert merely discloses that some traffic in the network 210 bound for the servers pass through the forwarding agent 231. However, “some traffic” disclosed in Albert does not teach or suggest the claimed “subsequent request” for the data source as recited in the claims. That is, any arbitrary “traffic”

or “data packet” disclosed in Albert does not teach or suggest the subsequent request recited in the independent claims.

Specifically, claim 1 expressly recites “subsequent request for the data source,” i.e., a request for the data source that is subsequent to the original request. Therefore, in order to disclose these features, the cited combination of the references would necessarily have to teach or suggest “subsequent request for the data source.”

In contrast, Albert at most discloses that it is not necessary to forward packets that are already in the flow and that some traffic in the network 210 bound for the servers passes through the forwarding agent 231. The “packets” and “some traffic” of Albert are not requests for the data source, as expressly recited in claim 1. As such, future packets are not subsequent requests for the data source. Accordingly, Albert does not teach or suggest “wherein the at least one port module bypasses the connection manager in the subsequent request.”

Yousefi’zadeh and Guenthner also do not teach or suggest “wherein the at least one port module bypasses the connection manager in the subsequent request” as recited in claims 1, 8 and 15.

The combination of Yousefi’zadeh, Guenthner and Albert do not teach or suggest that at least one port module sends an error message to the interface module indicating the unavailability of the data source

Claim 1 further recites, *inter alia*, “wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source, reestablishes a connection with the data source, and reconnects the remote application to the data source directly communicating with the remote application.” Claims 8 and 15 (in some variation) recite features similar to claim 1.

According to these features of claim 1, an error message is sent to the interface module indicating the unavailability of the data source from the port module.

In page 4 of the Office Action, the Examiner acknowledges that Yousefi'zadeh does not teach or suggest "wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source," but contends that FIG. 6 of Guenthner allegedly discloses these features.

Appellant respectfully disagrees for at least the following reasons.

In FIG. 6, Guenthner discloses a flowchart of Renew HAL (Hostname Address List) routine of the Hostname Process that provides an up-to-date HAL for use. However, FIG. 6 does not teach or suggest "wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source."

In particular, Guenthner discloses that the routine begins at step 92 by going through the HAL entries for the HAL returned. If the timestamp is older than Tx, then Status is set to "OK". At step 94, a test is performed to determine whether the list is a random list. If the outcome of the test at step 94 indicates that the list is not a random list, then the HAL is a primary/backup list. Further, if the outcome of a test at step 104 indicates that the list is older than Tz, or after step 108 for rebuilding HAL, the routine continues at step 110 to locate the first "OK" entry, which is then set to "Current." If no entry is "Current", the routine branches to step 112 and returns an error (column 7, lines 27-53). However, there is no teaching or suggestion of "wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source."

In the above discussed portion corresponding to FIGS. 6, Guentchner merely discloses that if no entry is "Current", then the routine returns an error. However, there is no teaching or suggestion of (i) an error message being sent to the interface module and (ii) an error message indicating the unavailability of the data source.

In the Advisory Action, the Examiner cites column 7, lines 40-53 of Guentchner for allegedly showing that an error message is returned. However, as noted above, this portion of Guentchner merely states that if no entry is "Current", the routine branches to step 112 and returns an error. However, there is no teaching or suggestion of where the error is returned, particular, does not teach or suggest "returning an error message to the interface module." That is, Guentchner does not teach or suggest where the error is returned in step 112. Therefore, Guentchner cannot and does not teach or suggest "wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source."

Yousefi'zadeh and Albert also do not teach or suggest the features of "wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source" as recited in claims 1, 8 and 15.

Thus, Appellant respectfully requests the Board to reverse the rejection of independent claims 1, 8 and 15.

2. Claims 6-7, 13-14, 18-22 and 24

Claims 6-7, 13-14, 18-22 and 24 that depend from claims 1, 8 and 15 are patentable over the cited combination of references at least by virtue of their dependency and the additional features recited therein.

B. CONCLUSION

In view of all the foregoing, Appellant respectfully submits that claims 1, 8 and 15, and the claims that depend therefrom, are not rendered unpatentable by the prior art. Accordingly, Appellant submits that all of the claims are in condition for immediate allowance and respectfully requests withdraw of the finality of the rejections and that the application be passed to issuance.

The fee required under 37 C.F.R. § 41.37(a) and 1.17(c) is being remitted. The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,

/Ebenesar D. Thomas/

SUGHRUE MION, PLLC
2100 Pennsylvania Avenue, N.W.
Washington, DC 20037
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

Ebenesar D. Thomas
Registration No. 62,499

Date: July 2, 2012

CLAIMS APPENDIX

CLAIMS 1, 6-8, 13-15, 18-22 and 24 ON APPEAL:

1. A method for automatically re-establishing a connection to a data source accessible by a plurality of remote applications, the method comprising:

providing at least one interface module configured to interface with a remote application;

providing at least one port module configured to interface between the interface module and the data source;

providing a connection manager to facilitate the interface between the interface module and the at least one port module;

detecting unavailability of the data source, by the at least one port module, in response to an initial request for the data source by the remote application;

dynamically detecting availability of the data source, by the at least one port module, in response to a subsequent request for the data source; and

re-connecting the data source to the remote application in response to the subsequent request,

wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source, reestablishes a connection with the data source, and reconnects the remote application to the data source directly communicating with the remote application, and

wherein the at least one port module bypasses the connection manager in the subsequent request.

6. The method of claim 1, wherein re-connecting further comprises re-establishing a connection between the at least one port module and the data source independently from initialization of the connection manager.

7. The method of claim 1, wherein re-connecting further comprises re-establishing a connection between the at least one port module and the data source without re-initializing the connection manager.

8. A non-transitory computer readable medium having stored thereon computer executable instructions for performing a method for connecting a plurality of remote applications with a data source, the method comprising:

providing at least one interface module configured to interface with a remote application;

providing at least one port module to interface between the interface module and the data source;

providing a connection manager to facilitate the interface between the interface module and the at least one port module;

detecting unavailability of the data source, by the at least one port module, in response to an initial request for the data source by the remote application;

dynamically detecting availability of the data source, by the at least one port module, in response to a subsequent request for the data source; and

re-connecting the data source to the remote application in response to the subsequent request,

wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source, reestablishes a connection with the data source, and reconnects the remote application to the data source by directly communicating with the remote application, and

wherein the at least one port module bypasses the connection manager in the subsequent request.

13. The computer readable medium of claim 8, wherein re-connecting further comprises re-establishing a connection between the at least one port module and the data source independently from initialization of the connection manager.

14. The computer readable medium of claim 8, wherein re-connecting further comprises re-establishing a connection between the at least one port module and the data source without re-initializing the connection manager.

15. A system for connecting a plurality of remote applications with a data source, the system comprising:

an interface module configured to interface with a remote application;
a port module, implemented by the processor, configured to interface between the interface module and the data source; and

a connection manager module configured to facilitate an interface between the interface module and the port module, wherein:

the port module further configured to detect unavailability of the data source in response to an initial request for the data source by the remote application, to dynamically detect availability of the data source in response to a subsequent request for the data source, and to re-connect the data source to the remote application in response to the subsequent request,

wherein the port module sends an error message to the interface module indicating the unavailability of the data source reestablishes a connection with the data source, and reconnects the remote application to the data source by directly communicating with the remote application, and

wherein the port module bypasses the connection manager in the subsequent request.

18. The system of claim 15, wherein re-connecting further comprises re-establishing a connection between the port module and the data source independently from initialization of the connection manager.

19. The system of claim 15, wherein re-connecting further comprises re-establishing a connection between the port module and the data source without re-initializing the connection manager.

20. The method of claim 1, further comprising connecting directly the interface module and the at least one port module for communicating independently from the connection manager in subsequent communications.

21. The computer readable medium of claim 8, further comprising connecting directly the interface module and the at least one port module for communicating independently from the connection manager in subsequent communications.

22. The system of claim 15, wherein the interface module and the port module are configured to be directly connected for communicating independently from the connection manager in subsequent communications.

24. The method of claim 1, wherein the port module reconnects the remote application to the data source without initialization by the connection manager.